

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-44356

(43) 公開日 平成9年(1997)2月14日

(51) Int.Cl. <sup>4</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/305			G 0 6 F 9/30	3 4 0 E
9/30	3 5 0			3 5 0 B

審査請求 未請求 請求項の数 7 O L (全 17 頁)

(21) 出願番号 特願平7-198609

(22) 出願日 平成7年(1995)8月3日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 藤川 義文

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 小島 啓二

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 西岡 清和

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(74) 代理人 弁理士 富田 和子

最終頁に続く

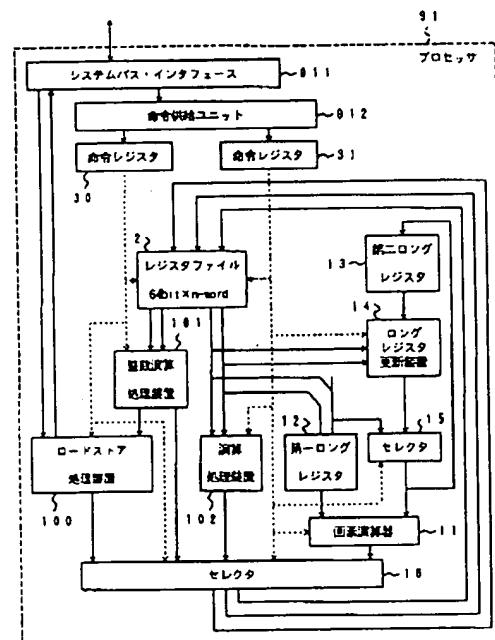
(54) 【発明の名称】 プロセッサおよびデータ処理装置

(57) 【要約】

【目的】データの一部を部分的に更新しながら行われる、レジスタファイルよりビット幅の大きいデータ間の演算を高速に行う。

【構成】レジスタファイル2と画素演算器11装置の間に、レジスタファイルのビット幅の2倍のビット幅を持つ第一ロングレジスタ12と第二ロングレジスタ13と、第二ロングレジスタ13のデータを部分的に更新するロングレジスタ更新装置14とを設ける。命令レジスタ31にロングレジスタ更新画素演算命令が格納されると、ロングレジスタ更新装置14は、第二ロングレジスタ13のデータの一部とレジスタファイル2から読み出したデータの一部を連結し、セクタ15を介して画素演算器11と第二ロングレジスタ13とに与える。画素演算器11は、第一ロングレジスタ12とセクタ15から与えられたデータ間の演算を行う。

図 2



## 【特許請求の範囲】

【請求項1】並列にデータを読みだし可能な少なくとも2つのレジスタを含んだ複数のビット幅Nのレジスタを備えたレジスタファイルを有するプロセッサであって、命令を格納する命令レジスタと、ビット幅M（但し、 $M > N$ ）のレジスタである第1のロングレジスタと、前記第1のロングレジスタから読みだしたデータの一部を前記レジスタファイルから読みだされたデータの一部に置き換えたMビット幅のデータを、前記命令レジスタに格納された命令に応じて生成する更新部と、前記レジスタファイルの複数のレジスタより並列に読みだされたデータを連結したMビット幅のデータ、もしくは、前記更新部が生成したMビット幅のデータの一方を、前記命令レジスタに格納された命令に応じて選択する選択部と、前記選択部が選択したMビット幅のデータに所定の演算を、前記命令レジスタに格納された命令に応じて施す演算器とを有し、前記第1のロングレジスタは、前記選択部が選択したMビット幅のデータを、前記命令レジスタに格納された命令に応じて格納することを特徴とするプロセッサ。

【請求項2】請求項1記載のプロセッサであって、前記レジスタファイルの複数のレジスタより並列に読みだされたデータを連結したMビット幅のデータを、前記命令レジスタに格納された命令に応じて格納するビット幅Mのレジスタである第2のロングレジスタを備え、前記演算器は、前記選択部が選択したMビット幅のデータと、前記第2のロングレジスタから読みだされたMビット幅のデータとの間に前記所定の演算を、前記命令レジスタに格納された命令に応じて施すことを特徴とするプロセッサ。

【請求項3】請求項2記載のプロセッサであって、前記命令レジスタに第1種の命令が格納された場合に、前記第2のロングレジスタは、前記レジスタファイルの複数のレジスタより並列に読みだされたデータを連結したMビット幅のデータを格納し、前記命令レジスタに第2種の命令が格納された場合に、前記選択部は前記レジスタファイルの複数のレジスタより並列に読みだされたデータを連結したMビット幅のデータを選択し、前記第1のロングレジスタは前記選択部が前記選択部が選択したMビット幅のデータを格納し、前記演算器は前記第2のロングレジスタから読みだされたMビット幅のデータと前記選択部が選択したMビット幅のデータとの間に所定の演算を施し、前記命令レジスタに第3種の命令が格納された場合に、前記更新部は、前記第1のロングレジスタから読みだしたデータの一部を前記レジスタファイルから読みだされたデータの一部に置き換えたMビット幅のデータを生成し、前記選択部は前記更新部が生成したMビット幅のデ

ータを選択し、前記第1のロングレジスタは前記選択部が選択したMビット幅のデータを格納し、前記演算器は前記第2のロングレジスタから読みだされたMビット幅のデータと前記選択部が選択したMビット幅のデータとの間に所定の演算を施すことを特徴とするプロセッサ。

【請求項4】請求項1記載のプロセッサであって、前記更新部は、前記レジスタファイルから読みだされたデータをシフトする第1のシフトと、前記第1のシフトでシフトされたデータと前記第1のロングレジスタから読みだしたデータとを連結したデータをシフトし、シフト後のデータ中の所定の範囲からMビット幅のデータを出力する第2のシフトとを備えていることを特徴とするプロセッサ。

【請求項5】請求項2記載のプロセッサであって前記演算器が施す演算は、前記選択部が選択したMビット幅のデータを複数の分割した複数の第1要素データの各々について、当該第1要素データと、前記第2のロングレジスタから読みだされたデータを複数の分割した複数の第2要素データのうちの前記Mビット幅のデータ中の位置に関して当該第1要素データと対応する第2要素データとの差の絶対値を求め、各第1要素データについて求めた差の絶対値の総和を求める演算であることを特徴とするプロセッサ。

【請求項6】請求項2記載のプロセッサであって前記演算器が施す演算は、前記選択部が選択したMビット幅のデータを複数の分割した複数の第1要素データの各々について、当該第1要素データと、前記第2のロングレジスタから読みだされたデータを複数の分割した複数の第2要素データのうちの前記Mビット幅のデータ中の位置に関して当該第1要素データと対応する第2要素データとの積を求め、各第1要素データについて求めた積の総和を求める演算であることを特徴とするプロセッサ。

【請求項7】複数の命令を含むプログラムとデータを記憶した記憶手段と、

前記記憶手段に記憶されたプログラム中の命令を順次取り込んで処理するプロセッサとを備えたデータ処理装置であって、

前記プロセッサは、

前記記憶手段から取り込んだ命令を格納する命令レジスタと、

並列にデータを読みだし可能な少なくとも2つのレジスタを含んだ複数のビット幅Nのレジスタを備えたレジスタファイルと、

前記レジスタファイルの各レジスタと前記記憶手段との間のデータの転送を、前記命令レジスタに格納された命令に応じて制御する手段と、

前記記憶手段から取り込んだ命令を格納する命令レジスタと、

ビット幅Mのレジスタである第1のロングレジスタと、

前記第1のロングレジスタから読みだしたデータの一部

を前記レジスタファイルから読みだされたデータの一部分に置き換えたMビット幅のデータを、前記命令レジスタに格納された命令に応じて生成する更新部と、  
前記レジスタファイルの複数のレジスタより並列に読みだされたデータを連結したMビット幅のデータ、もしくは、前記更新部が生成したMビット幅のデータの一方を、前記命令レジスタに格納された命令に応じて選択する選択部と、  
前記選択部が選択したMビット幅のデータに所定の演算を、前記命令レジスタに格納された命令に応じて施し、  
演算の結果を前記レジスタファイルのレジスタに書き込む演算器とを有し、  
前記第1のロングレジスタは、前記選択部が選択したMビット幅のデータを、前記命令レジスタに格納された命令に応じて格納することを特徴とするデータ処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ストアプログラム方式のプロセッサに関するものであり、特に、動画データ\*

$$f(u, v) = \sum_{j=0}^{15} \sum_{i=0}^{15} |A_{i,j} - B_{u+i, v+j}| \quad \dots (数1)$$

【0006】そして、このような評価式を用いる場合には、前記探索は、この評価式の値が最も小さくなる(u, v)の組合せを探すことにより成されることになる。

【0007】ここで、(数1)の評価式を高速に演算するための従来のプロセッサとしては、「MICROPROCESSOR※

$$d = \sum_{i=0}^7 |X_i - Y_i|$$

【0009】このプロセッサの構成の概要を図11に示す。

【0010】図示するように、このプロセッサは、ロード・ストア処理装置9130と(数2)に示した演算を行う画素演算処理装置9133とその他の演算を行う複数の演算装置9131、9132と、64ビット幅の複数のレジスタよりなるレジスタファイルと、それぞれの処理装置に対応して設けた命令レジスタ30～33と、その命令レジスタ30～33に命令を供給する命令供給ユニット912と、命令列を記憶した主記憶装置等が接続されているシステムバスとの入出力を制御するシステムバス・インタフェースなどから構成されている。

【0011】ここで、画素演算処理装置9133は、レジスタファイルから読み出されたデータを図12に示すように、8個の8ビットデータの集まりとして扱う。そして、レジスタファイルから読み出された2組みの8個の8ビットデータに対して、(数2)に示す演算を行う。

【0012】このようなプロセッサにおいて、画素演算★50

\*等を含むマルチメディアデータの処理に好適な構成に関するものである。

【0002】

【従来の技術】動画データを圧縮する方式としては、ISO/MPEGとして知られている方式がある。この方式では、動き補償を行うために、動画を構成する複数のフレーム間で、相互に画像が近似している部分を探索する。そして、この相互に近似している部分の位置の変化を、動画の動きを表す動きベクトルとして符号化することにより動画データの圧縮を行う。

【0003】このような探索は、図10のカレントフレーム80の中のカレントマクロブロックと呼ばれる16画素×16画素の領域の画像が、図1のリファレンスフレーム81の中のサーチウィンドウの中のどの部分に最も近似しているかを検索することによりより行われる。

【0004】ここで、画像間の近似の評価には、(数1)に示す評価式が広く用いられている。

【0005】

【数1】

※ REPORT, DECEMBER 5, 1994」の16ページに記載のSUN MicrosystemsのUltra SPARCが存在する。このプロセッサでは、(数2)に示す演算を1命令で行うことが出来る。

【0008】

【数2】

... (数2)

★処理装置9133に(数2)の演算を実行させる演算命令を用いて、(数1)を最小とする(u, v)の組合せを見つけ出す処理は図13のような手順で実現される。

【0013】すなわち、図13のステップ404は、ステップ402、409、410の働きによって、0-15の各vについて繰り返し、実行される。また、ステップ407も、同様に0-15の各vについて実行されるが、さらに、各vの値毎に、ステップ403、405、406の働きによって、1-15の間の各jの値について、繰り返し実行される。

【0014】次に、数3によって定義するr(j, u, v)を用いて説明すると、図13のステップ404、407では、上記の繰り返し処理の間に、0-15の間のuのそれぞれについて、0-15の間のjと0-15の間のvの各組合せ(j, u, v)について、r(j, u, v)を求める。ステップ404は、j=0についてr(j, u, v)を求め、ステップ407は1-15の間のjについてr(j, u, v)を求める。

【0015】

【数3】

$$r(j, u, v) = \sum_{i=0}^{15} |A_{i,j} - B_{u+i,v+j}| \quad \dots (数3)$$

【0016】また、これらのステップでは、上記処理の間に、同じ(u, v)の組合せに対して0-15の各jについて求めたr(j, u, v)の和を、Ru vとして求める。これは、vを固定し、ステップ404で0-15の間のuに対しj=0の場合のr(0, u, v)を求め、407で0-15の間のuに対し1-15の各jについてr(j, u, v)を求め、u vの組合せ毎に設けられるパラメータRu vに加算して処理を、0-15の間の各vについて行うことにより実現される。

【0017】そして、求まったRu vのうち先に求まったRu vより小さいもののみを残し(ステップ72)、最後に残ったRu vに対応する(u, v)の組合せを(数1)を最小とする(u, v)とする。ここで、(j, u, v)の特定の組合せに対するR(j, u, v)を算出するためには、(数2)に示す演算を2回、0-7のiについてと、8-15のiについてそれぞれ行う必要がある。結果、このプロセッサによれば、数2に示す演算と、この演算の前処理としてのデータの読みだしや、演算に用いるデータの作成を膨大な回数行う必要がある。

【0018】もちろん、前述したサーチウィンドウを狭くすることにより前記演算の回数を少なくして処理を高速化することもできるが、このようにすると圧縮効率が低下したり動画像の画質が劣化したりすることになる。

【0019】

【発明が解決しようとする課題】そこで、図11に示したプロセッサを、(数1)に示す演算を1命令で行えるように拡張することにより処理を高速化することが考えられる。

【0020】しかし、このためには、画素演算処理装置9133の入力ビット幅を広げると共に、レジスタファイルのビット幅も広げて、同時に多くの画素データを扱えるようにしなければならない。そして、このようにすると、レジスタファイルの規模が増大してしまう。また、画素演算処理装置9133以外の他の演算処理装置9131-9133では、これほどのビット幅のデータを必要としないので全体としてみて効率的な手法とは言えない。

【0021】また、レジスタファイルから読み出せるデータの個数を増やし、複数の画素演算処理装置において、並列に数2の演算を実行させるようにすることにより、処理を高速化することも考えられる。

【0022】しかし、このようにした場合にも、ハードウェア規模が大きく増大し、ハードウェアのコストが非常に大きくなる。

【0023】そこで、本発明は、ハードウェアのコスト\*

を、あまり増加することなく、レジスタファイルのデータ長より大きなデータ長のデータの演算処理を高速に実行できるプロセッサを提供することを目的とする。

【0024】

【課題を解決するための手段】前記目的達成のために、本発明は、並列にデータを読みだし可能な少なくとも2つのレジスタを含んだ複数のビット幅Nのレジスタを備えたレジスタファイルを有するプロセッサであって、命令を格納する命令レジスタと、ビット幅M(但し、M>N)のレジスタである第1のロングレジスタと、前記第1のロングレジスタから読みだされたデータの一部を前記レジスタファイルから読みだされたデータの一部に置き換えたMビット幅のデータを、前記命令レジスタに格納された命令に応じて生成する更新部と、前記レジスタファイルの複数のレジスタより並列に読みだされたデータを連結したMビット幅のデータ、もしくは、前記更新部が生成したMビット幅のデータの一方を、前記命令レジスタに格納された命令に応じて選択する選択部と、前記選択部が選択したMビット幅のデータに所定の演算を、前記命令レジスタに格納された命令に応じて施す演算器とを有し、前記第1のロングレジスタは、前記選択部が選択したMビット幅のデータを、前記命令レジスタに格納された命令に応じて格納することを特徴とするプロセッサを提供する。

【0025】また、このようなプロセッサであって、さらに、前記レジスタファイルの複数のレジスタより並列に読みだされたデータを連結したMビット幅のデータを、前記命令レジスタに格納された命令に応じて格納するビット幅Mのレジスタである第2のロングレジスタを備え、前記演算器は、前記選択部が選択したMビット幅のデータと、前記第2のロングレジスタから読みだされたMビット幅のデータとの間に前記所定の演算を、前記命令レジスタに格納された命令に応じて施すことを特徴とするプロセッサを提供する。

【0026】

【作用】本発明に係るプロセッサによれば、まず、たとえば、前記命令レジスタにある特定の命令が格納された場合に、前記選択部は前記レジスタファイルの複数のレジスタより並列に読みだされたデータを連結したMビット幅のデータを選択し、前記第1のロングレジスタは前記選択部が前記選択部が選択したMビット幅のデータを格納する。

【0027】そして、この後に前記命令レジスタに他の特定の命令が格納された場合には、前記更新部は、前記第1のロングレジスタから読みだされたデータの一部を前記レジスタファイルから読みだされたデータの一部に置

き換えたMビット幅のデータを生成し、前記選択部は前記更新部が生成したMビット幅のデータを選択し、前記第1のロングレジスタは前記選択部が選択したMビット幅のデータを格納し、前記演算器は前記前記選択部が選択したMビット幅のデータに所定の演算を施す。すなわち、前記他の特定の命令が与えられる度に、第1のロングレジスタに格納されたMビット幅のデータの一部をレジスタファイルから読みだしたデータの一部に置き換えることにより更新し、更新したMビットのデータに対して演算を施す。

【0028】したが、1命令に対して、レジスタファイルのビット幅よりビット幅の大きいデータの部分的な更新と、更新したデータの演算を行うことができ、処理が高速化される。また、レジスタファイルのビット幅は演算するデータのビット幅より小さくて済むのでハードウェアコストもあまり増加することがない。

【0029】

【実施例】以下、本発明の一実施例を説明する。

【0030】図1に、本実施例に係るプロセッサを適用した動画処理システムの構成を示す。

【0031】図中、91が本実施例に係るプロセッサであり、プロセッサ91はシステムバス92を介して主記憶装置93に接続している。また、同様に、プロセッサ91はシステムバス92を介して、二次記憶装置94や通信装置96やビデオ入出力装置95等や、その他のI/O装置97に接続している。主記憶装置93には、プログラムや、その他の作業データが記憶されている。また、主記憶装置93上には、画像を表すデータを記憶するフレームバッファ等も構成されている。ビデオ入出力装置95は、動画を撮影するビデオカメラ951や動画を表示するディスプレイ952との間でビデオ信号の入出力を担う。

【0032】このような構成の動画処理システムにおいて、ビデオカメラ951からビデオ入出力装置95によって取り込まれた動画データのデータは主記憶装置93内のフレームバッファに転送される。そして、主記憶装置93内の動画データはプロセッサ91によって処理され、その結果が二次記憶装置94に蓄えられたり、通信装置96によって、他のシステムに送られたりする。

【0033】次に、本実施例に係るプロセッサ91の内部構成を、図2に示す。

【0034】図示するように本実施例に係るプロセッサ91は、4つの読み出しポートと3つの書き込みポートを持つ64ビット幅のレジスタファイル2と、整数演算処理装置101と、他命令用演算処理装置102と、それぞれの演算処理装置に対応して設けた命令レジスタ30、31と、ロード・ストア処理装置100と、命令供給ユニット912、とシステムバス・インタフェース911とを備えている。また、図3に示すようにレジスタファイルのビット幅の2倍の128ビットからなる第一

ロングレジスタ12と第二ロングレジスタ13と、128ビット幅の2つの入力データを処理する画素演算器11と、第二ロングレジスタ13の内容を更新するロングレジスタ更新装置14設けられている。

【0035】レジスタファイル2は、たとえば、R0-R63の64個の64ビット幅のレジスタよりなり、4つの読み出しポートと3つの書き込みポートにより、4つのレジスタよりのデータの読みだしと3つのレジスタへの書き込みを同時に行うことができる。

10 【0036】さて、図1に示した主記憶装置93中のフレームバッファには、動画データとして、動画を構成するフレーム毎に、画素の輝度を表す1画素当たり8ビットの輝度データの配列が格納される。動画画像がカラーである場合には、動画データとして、画素の色成分を表す、たとえば1画素当たり8ビットの2種類の色データの配列も格納される。

20 【0037】主記憶装置93に格納された輝度データのうち演算に用いられる輝度データは、は、演算に先立ち、ロード・ストア処理装置100の制御下で、主記憶装置91より読みだされ、システムバスインタフェース91セクタ16を介してレジスタファイル2内に格納される。レジスタファイル2の各レジスタには、先に図12に示したように、8ビットの輝度データが格納される。

【0038】さて、図1の主記憶装置93に記憶されたプログラムの命令は、命令供給ユニット912によってシステムバス・インタフェース911を通し順次読み出され、命令の種類に応じて、図2の命令レジスタ30または命令レジスタ31に格納される。

30 【0039】命令レジスタ31内の命令に従って、レジスタファイル2の読み出しポートの内の2つと、他命令用演算処理装置101と、画素演算器11と、ロングレジスタ更新装置14と、画素演算ソースセクタ15、セクタ16と、レジスタファイル2の書き込みポートが制御される。

【0040】同様に、命令レジスタ30内の命令情報に従って、レジスタファイル2の読み出しポートの内の2つと、整数演算処理装置101と、ロード・ストア処理装置100と、セクタ16と、レジスタファイル2の書き込みポートが制御される。前述した輝度データのレジスタファイル2への主記憶装置93よりのロードも、この命令レジスタ30に格納された命令に従って、ロード・ストア処理装置100によって行われることになる。

【0041】ここで、命令レジスタ30に格納された一命令の実行と、命令レジスタ31に格納された一命令の実行は、並列的に同時に行うことができる。また、これらの命令の実行に伴うレジスタファイル2の読みだし及び書き込みを並列的に同時に行うことができる。

50 【0042】さて、このような構成において、本実施例

に係るプロセッサは、ロード命令やストア命令や加算命令次の3つの特徴的な命令をサポートする。すなわち、ロングレジスタ格納命令と、ロングレジスタ格納画素演算命令と、ロングレジスタ更新画素演算命令との3つの命令をサポートする。

【0043】これら命令は、命令供給ユニット912によって命令レジスタ31に格納される。

【0044】命令レジスタ31内にロングレジスタ格納命令が格納されると、命令レジスタ31の情報に従ってレジスタファイル2内の2つのレジスタが選択され、これら2つのレジスタに格納された64ビット幅のデータが2つ並列に2つの読みだしポートに出力される。そして、出力された2つのデータを連結した128ビット幅のデータが第一ロングレジスタ12内に格納される。

【0045】次に、命令レジスタ31内にロングレジスタ格納画素演算命令が格納されると、ロングレジスタ格納命令の場合と同様に、命令レジスタ31内の命令に従ってレジスタファイル2内の2つのレジスタが選択され、64ビット幅の2つのデータが読みだしポートに出力される。そして、出力された2つのデータを連結した128ビット幅のデータが、画素演算ソースセクタ15

に伝えられる。

【0046】画素演算ソースセクタ15は、連結された128ビット幅のデータを選択し出力する。画素演算\*

$$D = \sum_{i=0}^{15} |X_i - Y_i|$$

【0051】ただし、差演算器51は、入力する2つの画素演算ソース要素データの差の絶対値を計算する演算器であり、多入力加算器52は、それぞれの差演算器51の結果の総和を計算する演算器である。総和は、64ビット幅のデータとして出力される。本実施例では、65ビット目へのキャリアップは無視している。

【0052】次に、命令レジスタ31内にロングレジスタ更新画素演算命令が格納されると、命令レジスタ31の命令報に従ってレジスタファイル2内の1つのレジスタが選択され、64ビット幅のデータが1つの読みだしポートに出力され、ロングレジスタ更新装置14に伝えられる。また、命令レジスタ31の命令に含まれる位置情報とシフト情報がロングレジスタ更新装置14に伝えられる。また、これを並行して、ロングレジスタ更新装置14には、第二ロングレジスタ13の内容も伝えられる。

【0053】ここで、ロングレジスタ更新装置14の内部構成を図5に示す。

【0054】図示するように、ロングレジスタ更新装置14には、命令レジスタ31の命令に従って、レジスタファイル2の内の1つのレジスタから読みだされた64ビット幅のデータが第一ソースデータ線210を通し

て、連結元データ601として伝えられる。また、第二※50

\*ソースセクタ15から出力されたデータは、第二ロングレジスタ13と画素演算器11へ伝えられる。また、このとき第一ロングレジスタ12に格納されている128ビット幅のデータも、画素演算器11へ伝えられる。

【0047】第二ロングレジスタ13に伝えられた128ビット幅のデータは、第二ロングレジスタ13に格納される。一方、画素演算器11は、第一ロングレジスタ12から伝えられた128ビット幅のデータと画素演算ソースセクタ15から伝えられた128ビット幅のデータとの間で画素演算を行う。その画素演算の結果は、命令レジスタ31の命令で指定されたレジスタファイル2のレジスタに格納される。

【0048】ここで、画素演算器11の内部構成を図4に示しておく。

【0049】図示するように、この画素演算器11では、第一ロングレジスタ12と画素演算ソースセクタ15とから入力する2つの128ビットのデータである第一画素演算ソースデータ501と第二画素演算器ソースデータ502をそれぞれ、画素演算ソース要素データX0 5010のように、8ビット幅の数値データ16個の配列とみなし、(数4)に示す数式を演算するものである。

【0050】

【数4】

・・・(数4)

※ソースデータ線220を通して、命令レジスタ31内の命令に含まれている位置情報604とシフト情報605が伝えられる。

【0055】連結元データ601は、連結元データシフト61によって、位置情報604に従って左にシフトされる。位置情報604の数値をnとした場合のシフト量は8nビットである。これによって連結元データ601は、n要素データ(n画素の輝度データ)だけ左にシフトされ、シフト後に連結元データシフト61に残った64ビット幅のデータである連結元データシフト出力信号610として出力される。

【0056】連結元データシフト出力信号610の64ビット幅のデータと、第二ロングレジスタ出力データ線130を通して、第二ロングレジスタ13より入力した126ビット幅のデータである連結先データ602は、連結され、連結シフト62に伝えられる。連結シフト62は、連結された192ビット幅のデータを、シフト量情報605に従って左シフトし、ロングレジスタ更新データ線140上に、左側128ビットのデータをシフト結果として出力する。シフト量情報605の数値をnとした場合のシフト量は8nビットである。

【0057】このようなロングレジスタ更新装置14によって、第二ロングレジスタ13の内容を任意画素分だ

け左シフトし、右に空いた画素分を、第一ソースデータ線210上の任意の隣り合った画素で充填したデータを生成することが可能となる。

【0058】さて、ロングレジスタ更新装置14から出力された128ビット幅のデータ140（連結シフト62出力）は、ロングレジスタ更新データ線140を通して、画素演算ソースセクタ15に伝えられる。画素演算ソースセクタ15は、ロングレジスタ更新データ線140上のデータを選択し出力する。

【0059】画素演算ソースセクタ15からの出力データは、第二ロングレジスタ13と画素演算器11へ伝えられる。また、これと並行して、第一ロングレジスタ12に格納されているデータも、画素演算器11へ伝えられる。

【0060】第二ロングレジスタ13に伝えられたデータは第二ロングレジスタ13に格納される。

【0061】一方、画素演算器11は、前述したように、第一ロングレジスタ12から伝えられたデータと画素演算ソースセクタ15から伝えられたデータ間で演算を行う。その画素演算の結果は、命令レジスタ31の命令で指定されたレジスタファイル2のレジスタに格納される。

【0062】以上、本実施例に係るプロセッサ91がサポートする3つの命令が、どのように実行されるかを説明した。

【0063】さて、次に、このような3つの命令を用いて、先に図13に示した処理において、ステップ404やステップ407で行われる、特定の(j, v)の組合せに対して、数3のr(j, u, v)のu=0からu=15までの値を求める処理がどのように実現されるかを

説明する。

【0064】いま、特定の(j, v)の組合せに対して、数3のr(j, u, v)のu=0からu=15までの値を求める処理とは、図6に示す、サーチウィンドウの特定行から、順次左端となる画素を一つづつ右にずらしながら選択した16画素の組16個のそれぞれと、カレントフレームのカレントマクロブロックの特定行の16画素との輝度データとの差を求め、その総和を求める処理に相当する。

【0065】まず、この演算処理に先立ち、前述したように、レジスタファイル2のレジスタR20, R21にカレントフレームのカレントマクロブロックの特定行の16画素の輝度データを格納する。格納される輝度データと画素の関係は、図6に示す通りであり、左側の8画素をR20に、右側の8画素をR21に格納する。また、同様に、レジスタファイル2のレジスタR30-R33にサーチウィンドウの特定行の31画素の輝度データを格納する。格納される輝度データと画素の関係は、図6に示す通りであり、左側のものより8画素づつ順番に、R30, R31, . . . , R32の順に格納される。なお、サーチウィンドウの横のサイズは31画素であるため、R32の右端の8ビットは演算に用いられない画素の輝度データが格納される、もしくは、輝度データが格納ことになる。

【0066】次に、ロングレジスタ格納命令と、ロングレジスタ格納画素演算命令と、ロングレジスタ更新画素演算命令と、この処理で用いる加算命令のフォーマットを表1に示すように定める。

【0067】

【表1】

命 令	プログラムリスト中の表記	動 作
ロングレジスタ 格納命令	ldLR Rk, RI	$LR1 \leftarrow Rk // RI$
ロングレジスタ 格納 画素演算命令	ldLRdist Rj, Rk, RI	$LR2 \leftarrow Rk // RI$ $Rj \leftarrow \sum_{i=0}^{15}  LR1_i - LR2_i $
ロングレジスタ 更新 画素演算命令	updLRdist Rj, Rk, #l, #m	$LR2 \leftarrow (LR2_n // LR2_{n+1} // \dots // LR2_{15}$ $// Rk_l // Rk_{l+1} // \dots // Rk_{l+m-1})$ $Rj \leftarrow \sum_{i=0}^{15}  LR1_i - LR2_i $
画素演算命令	dist Rj, Rk, RI	$Rj \leftarrow \sum_{i=0}^7  Rk_i - RI_i $
画素配置命令	align Rj, Rk, RI, #m	$Rj \leftarrow (Rk_m // Rk_{m+1} // \dots // Rk_7$ $// RI_0 // RI_1 // \dots // Rk_{m-1})$
加算命令	add Rj, Rk, RI	$Rj \leftarrow Rk + RI$

ただし、LR1, LR2は、それぞれ第一/第二ロングレジスタを示すとする。  
 $Rn_i$ は、 $Rn$ のbit(8×i)…bit(8×i+7)の部分を示すとする。  
 $Rk//RI$ は、 $Rk$ と $RI$ の連結を示すとする。

【0068】ただし、表中のLR1は第一ロングレジスタ12を、LR2は第二ロングレジスタ13を表し、矢印は右辺の演算結果を左辺のレジスタに格納することを表す。また、矢印の右辺にある $Rn$ もしくは $LRn$ は、レジスタ $Rn$ もしくは $LRn$ に格納されたデータを表し、 $Rni$ もしくは $LRni$ は、レジスタ $Rn$ もしくは $LRni$ に格納されたデータの第(8×i)ビットから第(8×i+7)ビットの8ビットの輝度データを表すものとする。また、//は、//の両側の輝度データもしくは輝度データ列を連結することを表す。

【0069】また、ロングレジスタ更新画素命令の第3オペランドの#lが前述した位置情報604に、第4オペランドの#mが前述したシフト量情報605である。

【0070】このように命令のフォーマットを定めた場合、特定の(j, v)の組合せに対して、数3のr

(j, u, v)のu=0からu=15までの値を求める\*50

\*処理は、図7に示すプログラムによって実現することができる。

【0071】ただし、ロングレジスタ格納画素演算命令と、ロングレジスタ更新画素演算命令画素演算器11の処理はパイプライン処理が可能であり、そのレイテンシは3ステップであるとした。また、加算命令とロングレジスタ格納命令のレイテンシは1ステップであるとした。なお、図4の各差演算器51と多入力加算器52の間にラッチを挿入することにより、ロングレジスタ格納画素演算命令と、ロングレジスタ更新画素演算命令画素演算器11の処理のレイテンシは3となる。

【0072】また、スーパスカラまたはVL1Wアーキテクチャによって、命令レジスタ30に格納され整数演算処理装置101で実行される加算命令と、前述したように命令レジスタ31に格納され実行される他の命令は同時に実行できるものとして、同時実行可能な命令を



1行に記述した。

【0073】図7に示すように、このコーティング例では、20ステップでレジスタファイルのレジスタファイルR0からR15に特定の(j, v)の組合せに対して求めた、数3のr(j, u, v)のu=0からu=15までの値が格納されることになる。

【0074】このプログラムでは、右のコラムにおいて、第一行目でロングレジスタ格納命令idLRによって、第一ロングレジスタ12にレジスタファイル2のレジスタR20, R21に格納されているカレントフレームのカレントマクロブロックの特定行の16画素の輝度データを格納し、第二行目でロングレジスタ格納画素演算命令ldLRdistによって第一ロングレジスタ12に格納されたカレントマクロブロックのj行目の16画素の輝度データと、レジスタR30, R31に格納されたサーチウィンドウの特定行の左端の16画素の輝度データとの差の和を求めると共に、このサーチウィンドウの特定行の左端の16画素の輝度データを第二ロングレジスタ13に格納している。そして、第3行目から第17行目で、ロングレジスタ更新画素演算命令updLRdistによって、第二ロングレジスタ13に格納されている輝度データを8ビットシフトして第二ロングレジスタ13中の最も左の画素の輝度データを棄て、代わりに、第二ロングレジスタ13中の最も右の画素の右の画素の輝度データをレジスタファイルより読みだし連結したデータを作成し、これと、第2行目で第一ロングレジスタ12に格納されたカレントマクロブロックのj行目の16画素の輝度データとの差の和を求めている。

【0075】また、左側のコラムでは、加算命令addによって、第5行目から第29行目で右側のコラムの第2行目から第17行目で求めた和を順次加算している。画素演算器の処理のレイテンシは3ステップであるので、add命令は、初めに前記差の和を求める命令が発せられた第2行目の3ステップ後の第5行目から開始している。

【0076】ここで、参考までに、前述した従来のプロセッサ(図11参照)によって、特定の(j, v)の組合せに対して、数3のr(j, u, v)のu=0からu=15までの値を求める処理を実現するプログラムを図8に示す。

【0077】前掲した表1に示すように、図中の命令distは数2に示す演算を行う画素演算命令、alignは、レジスタファイルの2つのレジスタから読みだした2つの64ビット幅のデータを連結し、左にmビットシフトし、シフト後に左より64ビットをデータとして取り出す画素配置命令である。

【0078】ただし、画素演算命令distはパイプライン処理可能であり、そのレイテンシは2ステップであるとした。加算命令とロングレジスタ格納命令のレイテンシは1ステップであるとした。また、スーパーカラー

またはVLWアーキテクチャによって、異なる演算装置で実行される異なる種類の命令は同時に実行できるものとして、同時実行可能な命令を1行に記述した。

【0079】このプログラムでは、画素演算命令distでカレントフレームのカレントマクロブロックの特定行の16画素のうちの8画素の輝度データと、サーチウィンドウの特定行の8画素の輝度データとの間の差の和を求め、加算命令addで求めた和を順次加算し、画素配置命令alignで次の演算に用いるサーチウィンドウの特定行の8画素を右に1画素更新する処理を、サーチウィンドウの特定行の最も左側の8画素より16回繰り返して行う。

【0080】また、この処理と交互に、画素演算命令distでカレントフレームのカレントマクロブロックの特定行の16画素のうちの残りの8画素の輝度データと、サーチウィンドウの特定行の8画素の輝度データとの間の差の和を求め、加算命令addで求めた和を順次加算し、画素配置命令alignで次の演算に用いるサーチウィンドウの特定行の8画素を左に1画素更新する処理をサーチウィンドウの特定行の最も右側の8画素より16回繰り返して行う。

【0081】ただし、このプログラムは前述したUltraSPARCのものとは異なり、概略がわかる程度に簡略化したものである。

【0082】図8に示すように、このような命令によって前記処理を実現する場合には、49ステップを要している。

【0083】図7と図8の場合を比べると、本実施例に係るプロセッサの方が、この処理を約2.5倍高速に行うことができることになる。これは、本実施例に係るプロセッサの構成によれば、画素演算装置11において16画素の輝度データ間の演算を一度に行えることや、演算に用いるサーチウィンドウの16画素の輝度データの更新と、更新したデータと、カレントマクロブロックの16画素との間の演算を一命令で実現できることなどによるものである。

【0084】また、本実施例で用いた第一ロングレジスタ12、第二ロングレジスタ13、ロングレジスタ更新装置14、画素演算ソースセクタ15によって増加するハードウェア量は、レジスタファイルの読み出しポート数を増やしたり、レジスタファイルのビット幅を広げる場合のハードウェア増加量よりも少なく済む。

【0085】なお、以上の実施例では、画素演算器11を(数4)に示した演算を行うものとしたが、画素演算器11を他の演算を行うものとしてもよく、また、画素の演算ではなく一般のデータの演算を行うものとしてもよい。

【0086】たとえば、画素演算器11を(数5)に示す数式を演算するものとしてもよい。

【0087】

17

18

【数5】

$$D = \sum_{i=0}^{15} (X_i \times Y_i)$$

・・・ (数5)

【0088】この場合の画素演算器11の構成を図9に示す。図示するように、この構成は、先に図4に示した構成における差演算器51を、乗算器53に置き換えた構成を有している。乗算器53は、2つの要素データの積を計算する演算器である。

\* 【0089】このような画素演算器11は、(数6)で表されるような時系列データのフィルタ演算などに応用できる。

【0090】

【数6】

$$Y_t = \sum_{i=0}^{15} (w_i \times x_{i+t})$$

・・・ (数6)

【0091】このような応用によれば、タップ数が多い、したがって、良好な応答特性を得ることができ、かつ、処理速度の高速なフィルタを実現することができる。

【0092】

【発明の効果】以上説明してきたように、本発明によれば、ハードウェアのコストを、あまり増加することなく、レジスタファイルのデータ長より大きなデータ長のデータ間の演算処理を高速に実行できるプロセッサを提供することができる。

【図面の簡単な説明】

【図1】本発明の実施例に係る動画処理システムの構成を示すブロック図である。

【図2】本発明の実施例に係るプロセッサの構成を示すブロック図である。

【図3】本発明の実施例に係る第一、第二ロングレジスタを示す図である。

【図4】本発明の実施例に係る画素演算器の構成を示すブロック図である。

【図5】本発明の実施例に係るロングレジスタ更新装置の構成を示すブロック図である。

【図6】本発明の実施例におけるレジスタファイルへの輝度データの格納のようすを示した図である。

【図7】本発明の実施例に係るプロセッサのプログラム例を示した図である。

【図8】従来のプロセッサのプログラム例を示した図である。

【図9】本発明の実施例に係る画素演算器の他の構成例を示したブロック図である。

【図10】動画データ圧縮処理で行われるフレーム間の近似画素ブロックの探索範囲を示した図である。

※【図11】従来のプロセッサの構成を示したブロック図である。

【図12】従来のプロセッサにおけるレジスタファイルへの輝度データの格納のようすを示した図である。

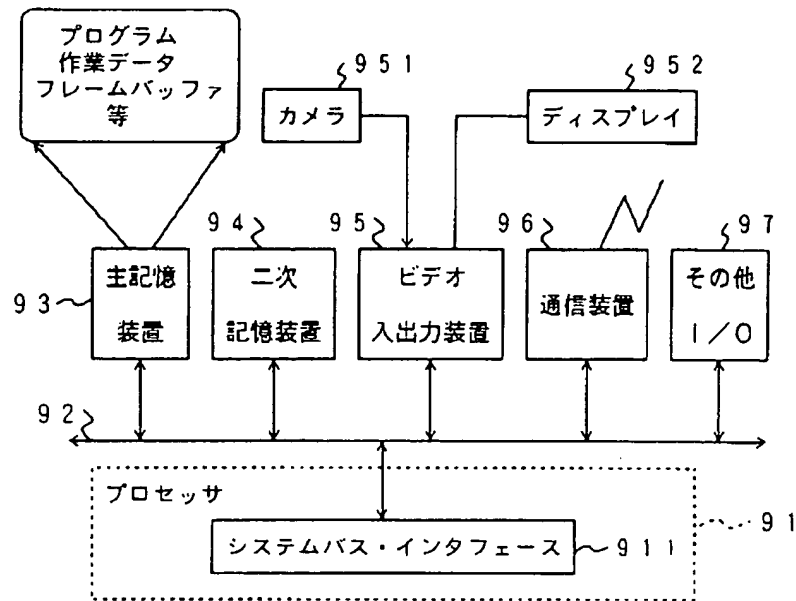
【図13】従来のプロセッサにおける近似画素ブロック探索の手順を示したフローチャートである。

【符号の説明】

2	レジスタファイル
11	画素演算器
12	第一ロングレジスタ
13	第二ロングレジスタ
14	ロングレジスタ更新装置
15	画素演算ソースセクタ
16	レジスタファイル書き込みセクタ
30	命令レジスタ
31	命令レジスタ
91	プロセッサ
92	システムバス
93	主記憶装置
94	二次記憶装置
95	ビデオ入出力装置
96	通信装置
97	I/O装置
100	ロード・ストア処理装置
101	整数演算処理装置
102	他命令用演算処理装置
911	システムバス・インタフェース
912	命令供給ユニット
951	カメラ
952	ディスプレイ

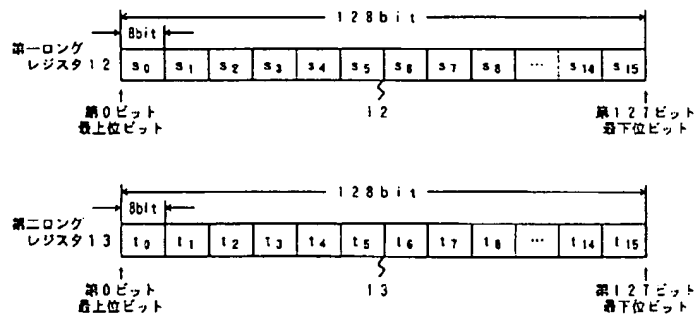
【図1】

図 1



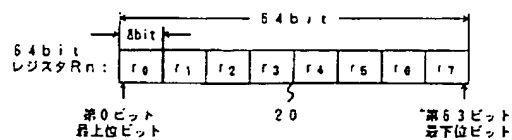
【図3】

図 3



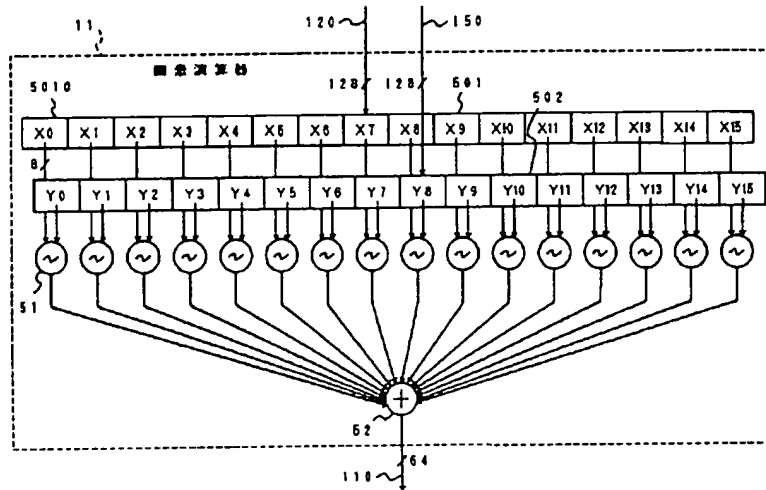
【図12】

図 12



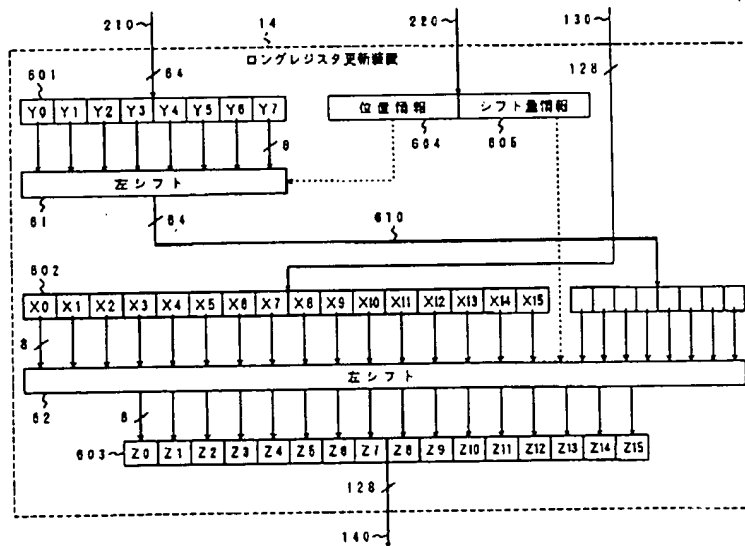
【図4】

図 4

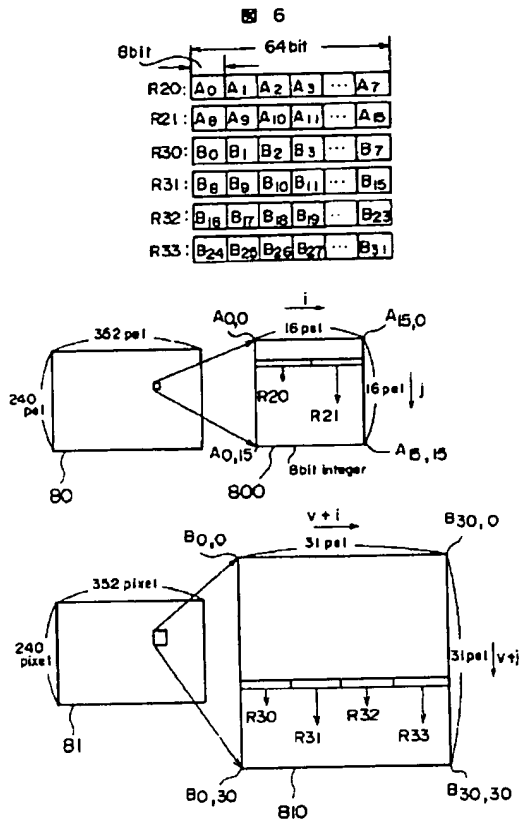


【図5】

図 5



【図6】



【図7】

図7は、レジスタ操作のリストを示す。

アドレス	操作	レジスタ
001:	-	ldLR R20, R21
002:	-	ldLRdist R40, R30, R31
003:	-	updLRdist R41, R32, #0, #1
004:	-	updLRdist R42, R32, #2, #1
005:	add R0, R0, R40	updLRdist R43, R32, #2, #1
006:	add R1, R1, R41	updLRdist R40, R32, #3, #1
007:	add R2, R2, R42	updLRdist R41, R32, #4, #1
008:	add R3, R3, R43	updLRdist R42, R32, #5, #1
009:	add R4, R4, R40	updLRdist R43, R32, #6, #1
010:	add R5, R5, R41	updLRdist R40, R32, #7, #1
011:	add R6, R6, R42	updLRdist R41, R33, #0, #1
012:	add R7, R7, R43	updLRdist R42, R33, #1, #1
013:	add R8, R8, R40	updLRdist R43, R33, #2, #1
014:	add R9, R9, R41	updLRdist R40, R33, #3, #1
015:	add R10, R10, R42	updLRdist R41, R33, #4, #1
016:	add R11, R11, R43	updLRdist R42, R33, #5, #1
017:	add R12, R12, R40	updLRdist R43, R33, #6, #1
018:	add R13, R13, R41	-
019:	add R14, R14, R42	-
020:	add R15, R15, R43	-

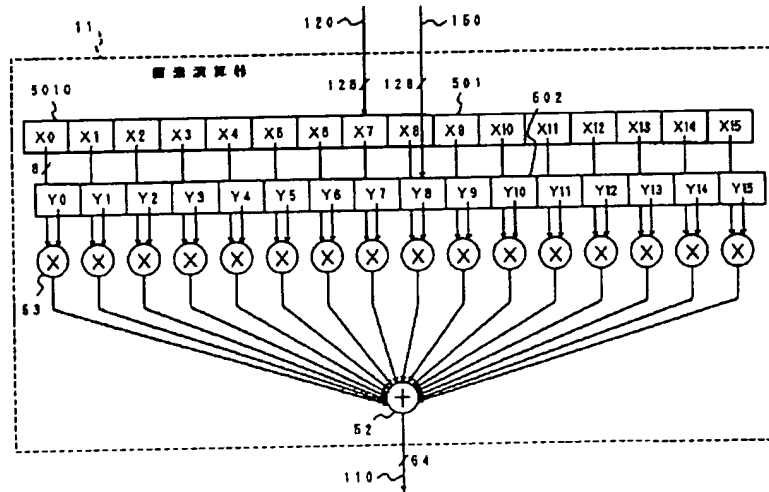
【図8】

図8は、レジスタ操作のリストを示す。

アドレス	操作	レジスタ
001:	-	dist R40, R20, R30
002:	align R34, R30, R31, #1	dist R41, R21, R31
003:	add R0, R0, R40	dist R35, R31, R32, #1
004:	add R0, R0, R41	dist R42, R20, R34
005:	align R34, R30, R31, #2	dist R43, R21, R35
008:	add R1, R1, R42	align R35, R31, R32, #2
007:	add R1, R1, R43	dist R40, R20, R34
008:	align R34, R30, R31, #3	dist R41, R21, R35
009:	add R2, R2, R40	dist R35, R31, R32, #3
010:	add R2, R2, R41	dist R42, R20, R34
011:	align R34, R30, R31, #4	dist R43, R21, R35
012:	add R3, R3, R42	align R35, R31, R32, #4
013:	add R3, R3, R43	dist R40, R20, R34
014:	align R34, R30, R31, #5	dist R41, R21, R35
015:	add R4, R4, R40	align R35, R31, R32, #5
016:	add R4, R4, R41	dist R42, R20, R34
017:	align R34, R30, R31, #6	dist R43, R21, R35
018:	add R5, R5, R42	align R35, R31, R32, #6
019:	add R5, R5, R43	dist R40, R20, R34
020:	align R34, R30, R31, #7	dist R41, R21, R35
021:	add R6, R6, R40	dist R35, R31, R32, #7
022:	add R6, R6, R41	dist R42, R20, R34
023:	-	dist R43, R21, R35
024:	add R7, R7, R42	align -
025:	add R7, R7, R43	dist R40, R20, R31
026:	align R34, R31, R32, #1	dist R41, R21, R32
027:	add R8, R8, R40	dist R35, R32, R33, #1
028:	add R8, R8, R41	dist R42, R20, R34
029:	align R34, R31, R32, #2	dist R43, R21, R35
030:	add R9, R9, R42	align R35, R32, R33, #2
031:	add R9, R9, R43	dist R40, R20, R34
032:	align R34, R31, R32, #3	dist R41, R21, R35
033:	add R10, R10, R40	align R35, R32, R33, #3
034:	add R10, R10, R41	dist R42, R20, R34
035:	align R34, R31, R32, #4	dist R43, R21, R35
036:	add R11, R11, R42	align R35, R32, R33, #4
037:	add R11, R11, R43	dist R40, R20, R34
038:	align R34, R31, R32, #5	dist R41, R21, R35
039:	add R12, R12, R40	align R35, R32, R33, #5
040:	add R12, R12, R41	dist R42, R20, R34
041:	align R34, R31, R32, #6	dist R43, R21, R35
042:	add R13, R13, R42	align R35, R32, R33, #6
043:	add R13, R13, R43	dist R40, R20, R34
044:	align R34, R31, R32, #7	dist R41, R21, R35
045:	add R14, R14, R40	align R35, R32, R33, #7
046:	add R14, R14, R41	dist R42, R20, R34
047:	-	dist R43, R21, R35
048:	add R15, R15, R42	-
049:	add R15, R15, R43	-

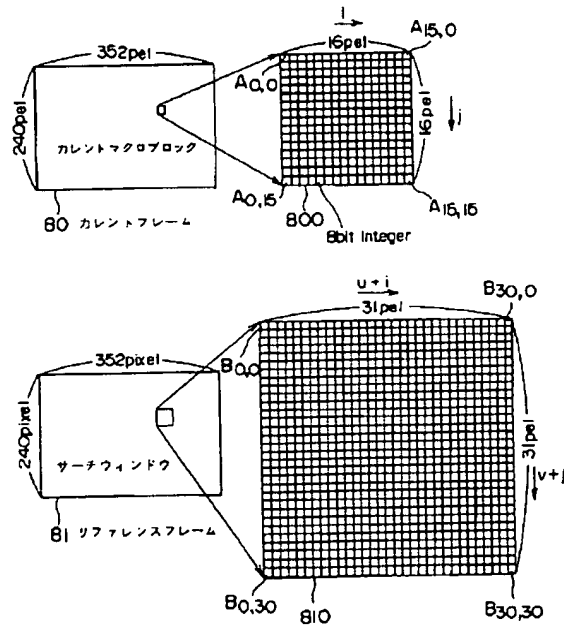
【図9】

図 9



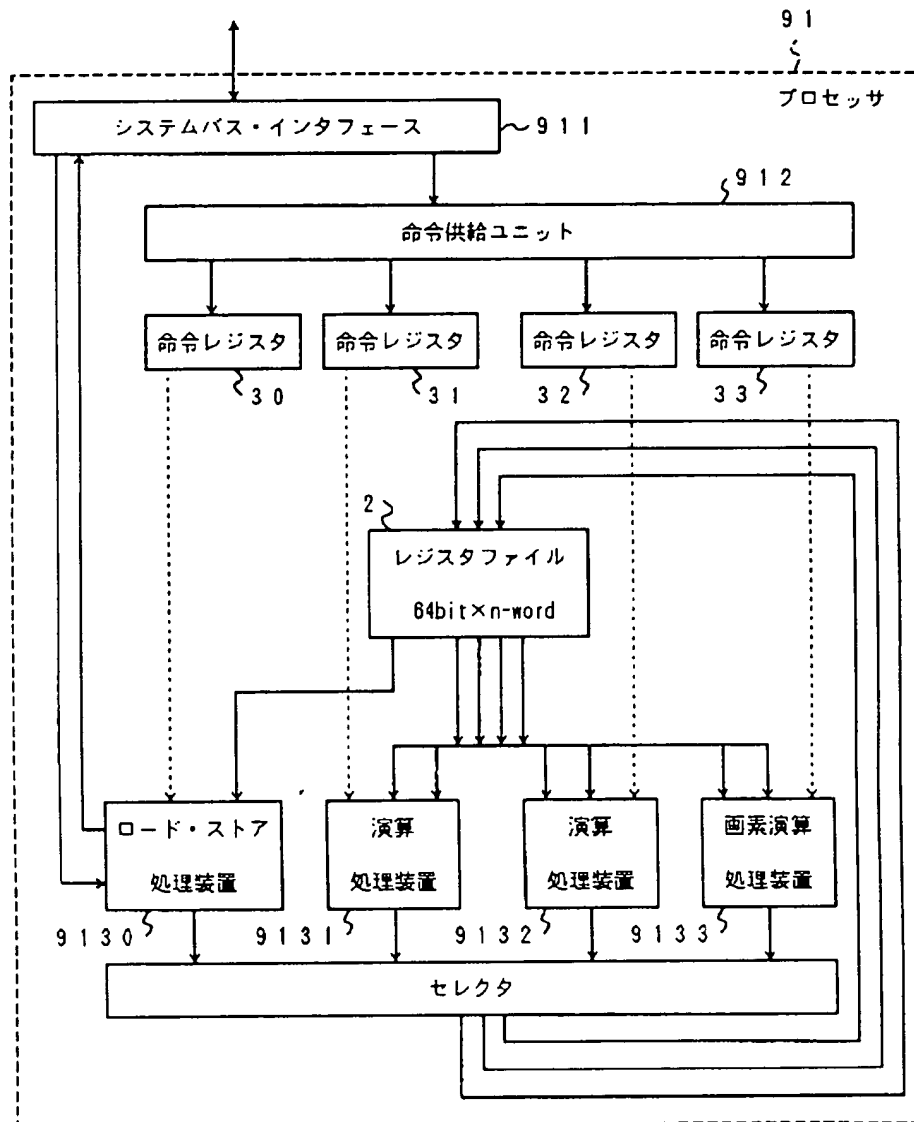
【図10】

図 10



【図11】

図11



DERWENT-ACC-NO: 1997-184311  
DERWENT-WEEK: 200104  
COPYRIGHT 1999 DERWENT INFORMATION LTD

TITLE: Data processor for multimedia data especially moving image data - has calculator that performs predetermined operation to selected data of bits with M face widths based on instruction stored in instruction register

INVENTOR: FUJIKAWA, Y; ISHIGURO, M ; KOJIMA, K ; NISHIOKA, K ; NOJIRI, T ; TANAKA, K

PATENT-ASSIGNEE: HITACHI LTD[HITA], HITACHI SEISAKUSHO KK[HITA]

PRIORITY-DATA: 1995JP-0198609 (August 3, 1995)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
KR 213604 B1	August 2, 1999	N/A	000	G06F 009/34
JP 09044356 A	February 14, 1997	N/A	017	G06F 009/305
TW 297878 A	February 11, 1997	N/A	000	G06F 015/76
KR 97014366 A	March 29, 1997	N/A	000	H04N 007/26
US 5870618 A	February 9, 1999	N/A	000	G06F 009/26

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO	APPL-DATE
KR 213604B1	N/A	1996KR-0030919	July 29, 1996
JP 09044356A	N/A	1995JP-0198609	August 3, 1995
TW 297878A	N/A	1996TW-0109033	July 24, 1996
KR 97014366A	N/A	1996KR-0030919	July 29, 1996
US 5870618A	N/A	1996US-0681180	July 22, 1996

INT-CL\_(IPC): G06F009/26; G06F009/30 ; G06F009/305 ; G06F009/34 ; G06F015/76 ; H04N007/26

ABSTRACTED-PUB-NO: JP 09044356A

BASIC-ABSTRACT: The processor (91) has a register file (2) and at least two long registers (12,13). The long registers can read data in parallel and stores several bits with face widths of M and N, respectively, in which M is greater than N. Instruction registers (30,31) are provided to store instructions. An updating unit (14) forms data for the bits with M face widths which transposed a portion of the data read from the first long register into a portion of the data read from the register file, based on the stored instruction.



A selector (15) chooses either the data of the bits with M face widths which parallelly coupled the read data from the register file to the data from the first long register, or the formed data from the updating unit. A calculator (11) performs a predetermined operation to the selected data of the bits with M face widths based on the stored instruction. The first long register stores the selected data of the bits with M face widths based on the instruction stored in the instruction register.

ADVANTAGE - Performs high-speed calculation to data with longer lengths than data in register file without increasing hardware cost.

ABSTRACTED-PUB-NO: US 5870618A

EQUIVALENT-ABSTRACTS: The processor (91) has a register file (2) and at least two long registers (12,13). The long registers can read data in parallel and stores several bits with face widths of M and N, respectively, in which M is greater than N. Instruction registers (30,31) are provided to store instructions. An updating unit (14) forms data for the bits with M face widths which transposed a portion of the data read from the first long register into a portion of the data read from the register file, based on the stored instruction.

A selector (15) chooses either the data of the bits with M face widths which parallelly coupled the read data from the register file to the data from the first long register, or the formed data from the updating unit. A calculator (11) performs a predetermined operation to the selected data of the bits with M face widths based on the stored instruction. The first long register stores the selected data of the bits with M face widths

based on the instruction stored in the instruction register.

ADVANTAGE - Performs high-speed calculation to data with longer lengths than data in register file without increasing hardware cost.

CHOSEN-DRAWING: Dwg.2/13

TITLE-TERMS:

DATA PROCESSOR DATA MOVE IMAGE DATA CALCULATE PERFORMANCE  
PREDETERMINED OPERATE  
SELECT DATA BIT FACE WIDTH BASED INSTRUCTION STORAGE INSTRUCTION  
REGISTER

DERWENT-CLASS: T01

EPI-CODES: T01-J30;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N1997-151834